Data Science in R (DSR) – Testat 1

Markus Gumbel

12. Oktober 2026

Inhaltsverzeichnis

Hinweise **1**

1 Eigenes Histogramm **3**
1.1 Funktion `myhistogram` . . . 3
1.2 Beispieldaten . . . 3
1.3 Barplot . . . 4

2 Visualisierung von Datensätzen **5**
2.1 Säuglingssterblichkeit und Wirtschaftsleistung . . . 5
2.2 Kieferwachstum bei Kindern . . . 6

3 Krebsfälle in Deutschland **7**
3.1 Einlesen der Daten . . . 7
3.2 Übersicht aller Krebsarten . . . 7
3.3 Prävalenz nach Altergruppen und Geschlecht . . . 7

Hinweise

Aktion		Datum	Ablauf
Ausgabe	Mo.,	12.10.26	über Moodle
Abgabe	Mo.,	02.11.25, 09:00 Uhr	über Moodle
Testat	Mo.,	02.11.26	als Präsentation

- Die Abnahme der Testate gilt als Prüfungsleistung. Bei einer Verhinderung durch Krankheit ist eine ärztliche Bescheinigung der Arbeitsunfähigkeit vorzulegen.
- Geben Sie Ihre Lösungen fristgerecht via Moodle ab. Es werden dort folgende Abgaben erwartet (Ausarbeitungen in einem anderen Format werden nicht berücksichtigt):
  - Für jede Aufgabe ein R-Notebook (als *.Rmd-Datei) und etwaige benötigten R-Skripte (*.R).
  - Genau ein PDF mit allen Lösungen, das außerdem die Gruppennummer und alle Namen enthält.
- Nutzen Sie das R-Notebook, das es für jede Aufgabe gibt und **fügen Sie dort Ihre Lösungen ein**, d.h. Source-Code und normaler Text.
- **Kommentieren Sie Ihre Lösungen ausführlich** – entweder als normalen Text im Notebook oder als Kommentar in R-Anweisungen. Ohne Kommentare gibt es Punktabzug.
- Jedes abgegebene R-Notebook muss ohne Fehler auf einem anderen Rechner mit R (4.1 oder höher) ausgeführt werden können und ein HTML-Dokument erzeugen. Installations-

anweisungen für die benutzen Packages sind nicht nötig – laden Sie aber die Packages im Notebook.
- Während der Abnahme sind die Ergebnisse am Rechner live zu demonstrieren.
- Bei der Abnahme der Übung ist der Studentenausweis vorzulegen.

1 Eigenes Histogramm

Es soll ein eigenes Histogramm erzeugt werden. Der Dateiname für das Skript ist `myhistogram.R`.

1.1 Funktion `myhistogram`

Programmieren Sie in R die Funktion `myhistogram`, die als Parameter `x` einen Vektor aus Zahlen erhält. Die Zahlen werden in `n` Intervalle einsortiert, und es wird gezählt, wie oft eine Zahl in einem Intervall vorkommt. Der Rückgabewert ist eine Liste mit den Einträgen `borders`, die die $n+1$ Intervallgrenzen enthalten und `counts`, der die Anzahlen enthält.

- Die n Intervalle sollen gleich groß sein (Δb), d.h. für die Intervallgrenzen $b_1, b_2, \ldots, b_{n+1}$ gilt $\frac{b_{i+1}-b_i}{n} = \Delta b$ für $i = 1, 2, \ldots n$.
- Die äußeren Grenzen b_1 und b_n sollen als optionale Parametern `min` und `max` an die Funktion übergeben werden. Werte aus `x`, die zu keinem Intervall gehören, sollen ignoriert werden. Es wird aber eine Warnung ausgegeben, die sagt, welche Zahlen außerhalb des Bereichs liegen.
- Eine Zahl z gehört zum i-ten Intervall, falls $b_i \leq z < b_{i+1}$ gilt.

Bis auf `x` sollen alle Parameter optional sein. Überlegen Sie sinnvolle Default-Werte.

Es ist natürlich **nicht** erlaubt, in der eigenen Funktion andere Funktionen zu nutzen, die ein Histogramm erzeugen.

Hier ein Beispiel:

```
# Bitte mit echter Lösung in myhistogram.R ersetzen:
myhistogram = function(x) {
  list(borders = c(), counts = c())
}
x = seq(-5, 6, by = 1 / 3)
l = myhistogram(x)
print(l$borders)
print(l$counts)
```

1.2 Beispieldaten

Hier zunächst zwei Beispiele.

1.2.1 Beispiel 1

Es wird eine Warnung ausgeben:

```
## Warning in myhistogram(x, n = 10, min = -5, max = 6): Zahl(en) außerhalb
## Intervallgrenzen: 6
```

```
x = seq(-5, 6, by = 1 / 3)
# myhistogram(x, n = 10, min = -5, max = 6)
solution = list(
  borders = c(-5.0, -3.9, -2.8, -1.7, -0.6, 0.5, 1.6, 2.7, 3.8, 4.9, 6.0),
  counts = c(4, 3, 3, 4, 3, 3, 4, 3, 3, 3)
)
```

1.2.2 Beispiel 2

```
x = seq(-5, 6, by = 1 / 3)
# myhistogram(x, n = 5, min = -10, max = 10)
solution = list(borders = c(-10, -6, -2, 2, 6, 10),
                counts = c(0, 9, 12, 12, 1))
```

1.2.3 Beispiel 3

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

1.2.4 Beispiel 4

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

1.2.5 Beispiel 5

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

1.3 Barplot

Nutzen Sie Ihre Funktion `myhistogram` und erzeugen Sie einen Barplot mit *ggplot*. Die x-Achse zeigt dabei die Mitte des Intervalls und die y-Achse die Anzahl der Elemente in dieser Klasse.

Tipp: Der Parameter `stat` von `geom_bar` ist wichtig.

Vervollständigen Sie den Chunk. Die Kommentare sollen zu Anweisungen umgewandelt werden:

```
set.seed(1)
x = rnorm(0, 1, n = 1000)
# h = myhistogram(x, n = 20)
# ggplot() + ...
```

2 Visualisierung von Datensätzen

In diesem Abschnitt sollen alle Graphiken mit *ggplot* und alle Tabellen mit `kable` erstellt werden.

2.1 Säuglingssterblichkeit und Wirtschaftsleistung

Nutzen Sie den Datensatz `carData::UN`. In der Hilfe finden Sie Hinweise, was dort gezeigt ist. Beachten Sie, dass für einige Länder Werte fehlen (`NA`); diese Länder sollen nicht berücksichtigt werden.

2.1.1 Wirtschaftsleistung vs. Säuglingssterblichkeit

Erzeugen Sie diese Graphik, indem Sie den nachfolgenden Chunk vervollständigen. Die gezeigten Länder sind Germany, United States, China, India, Afghanistan, Sierra Leone, Japan, Cuba.

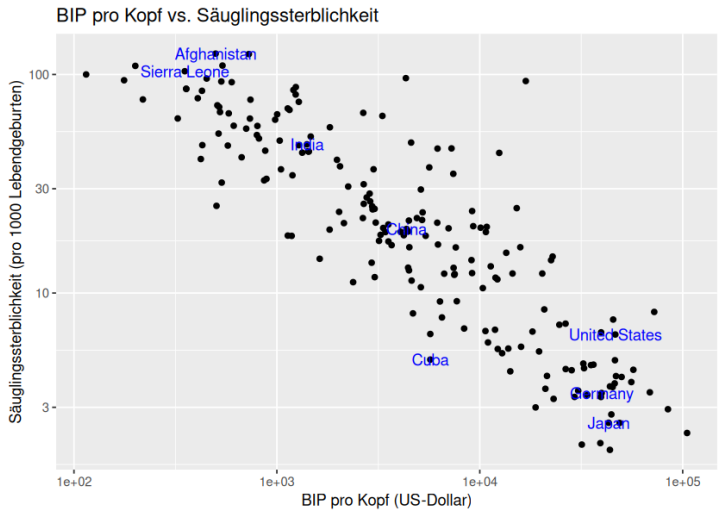


Abbildung 1: BIP pro Kopf vs. Säuglingssterblichkeit.

Tipp: Sie dürfen (und sollen) weitere Libraries nutzen, wenn diese hilfreich sind.

```
# Ihre Lösung:
```

2.1.2 Säuglingssterblichkeit im Verhältnis zur Wirtschaftsleistung

Geben Sie diejenigen 10 Länder als Tabelle im Notebook aus, die das größte Verhältnis r von Säuglingssterblichkeit zu Wirtschaftsleistung haben. Dabei ist r die Säuglingssterblichkeit (pro 1000 Lebendgeburten) je 1000 US-Dollar BIP pro Kopf. Die Liste soll nach r absteigend sortiert sein und den Ländernamen und r enthalten.

Vervollständigen Sie diesen Chunk:

```
# Ihre Lösung:
```

Geben Sie nun – wie eben – diejenigen 10 Länder als Tabelle aus, die das **kleinste** Verhältnis r haben. Die Liste soll nach r absteigend sortiert sein.

Vervollständigen Sie diesen Chunk:

```
# Ihre Lösung:
```

2.2 Kieferwachstum bei Kindern

Nutzen Sie den Datensatz `nlme::Orthodont`. In der Hilfe finden Sie Hinweise, was dort gezeigt ist.

2.2.1 Überblick über Verlauf bei allen Kindern

Plotten Sie im folgenden Chunk den Verlauf des Abstands zwischen Hypophyse und Fossa pterygopalatina (y-Achse) in Abhängigkeit vom Alter (x-Achse). Dies soll in einem Diagramm mit Unterdiagrammen erfolgen: ein Unterdiagramm zeigt den Verlauf für je ein Kind (Subject) und dessen Geschlecht (Male oder Female).

```
# Ihre Lösung:
```

2.2.2 Boxplots des Abstands je Alter

Erzeugen Sie ein Diagramm, das in zwei Unterdiagrammen für die Jungen und die Mädchen Boxplots erstellt. Die Boxplots geben die Verteilung des Abstands je Alter an. In Anlehnung an das obige Diagramm sollen die Boxplots vertikal ausgerichtet sein.

```
# Ihre Lösung:
```

3 Krebsfälle in Deutschland

In dieser Aufgabe sollen Daten zu Krebsfällen in Deutschland visualisiert werden.

3.1 Einlesen der Daten

Auf der Webseite

- https://www.krebsdaten.de/

gibt es unter "Datenbankabfrage" die Möglichkeit, Daten zu Krebsfällen in Deutschland herunter zu laden. Wählen Sie *Prävalenz* sowie *Fallzahlen* als Datenquelle aus. Die Prävalenz beschreibt in der Epidemiologie und medizinischen Statistik z.B. die Krankheitshäufigkeit innerhalb der Bevölkerung.

Exportieren Sie **alle** verfügbaren Daten als CSV-Datei bis auf das Attribut *"Intervall-Länge in Jahren"*. Hier soll nur die Intervall-Länge 5 berücksichtigt werden.

Es bleibt Ihnen überlassen, in *welcher Struktur* Sie die Daten exportieren. **Tipp**: Schauen Sie in den weiteren Unteraufgaben nach, um zu entscheiden, welches Format sinnvoll sein könnte.

Wichtig: Es darf in dieser Aufgabe nur **eine** Datei exportiert werden.

Lesen Sie die CSV-Datei ein und bereiten Sie einen Data-Frame vor, der für die späteren Visualisierungen genutzt werden kann. Achten Sie darauf, dass die Datentypen der Spalten sinnvoll sind. **Tipp**: Der Funktionsaufruf `as.numeric(gsub("\\.", "", "123.567"))` erzeugt die Zahl 123567. Das ist nützlich, wenn Zahlen einen Tausender-Trenner enthalten.

Geben Sie die ersten 10 Zeilen und 7 Spalten als Tabelle aus.

```
# Ihre Lösung:
```

3.2 Übersicht aller Krebsarten

Erzeugen Sie ein Balkendiagramm, das für das jüngste zur Verfügung stehende Jahr (z.B. 2019) eine Übersicht über die Prävalenz aller Krebsarten für die Gruppe der 55- bis 64-Jährigen gibt. Hierbei soll pro Krebsart das Geschlecht unterschieden werden. Es sollen **nicht** mehrere Balkendiagramme dafür erstellt werden - nur genau eins. Es bleibt Ihnen überlassen, wie Sie die Balken darstellen. Begründen Sie aber kurz Ihre Motivation.

Weitere Anforderungen:

- verständliche Label für die Achsen
- Tabellenüberschrift

```
# Ihre Lösung:
```

3.3 Prävalenz nach Altergruppen und Geschlecht

Erzeugen Sie für den zur Verfügung stehenden Zeitraum (z.B. 2004 bis 2019) eine Graphik, die für jedes Jahr die Prävalenz der gesamten Krebsfälle anzeigt. Hierbei sollen das Geschlecht und zwei Alterstufen unterschieden werden. Die erste Alterstufe umfasst alle Patienten bis zu 54 Jahren und die zweite alle ab 55 Jahren.

Die Graphik soll in einer Untergraphik die Männer links und die Frauen rechts zeigen. Jede Untergraphik zeigt die Kurven für die beiden Altersgruppen.

Weitere Anforderungen:

- verständliche Label für die Achsen
- Tabellenüberschrift
- Die Prävalenz soll auf einer Skala angezeigt werden, die bei 0 beginnt, damit die Daten besser vergleichbar sind.

```
# Ihre Lösung:
```